

Bayesian statistics and some model choice

gean piere ventura cruz

2022-10-17

Task 1

We are still using the functions from the “Likelihood, numerical optimization and the Bootstrap” project. We are going to compute AIC for the model using functions from the “Score-, Wald- and likelihood ratio tests” project and make sure it agrees with R’s value. Also compute the corresponding value based on leave-one-out cross validation

```
summary(modell)
```

```
##
## Call:
## glm(formula = Resultat ~ Alder + Kon + Utbildare, family = "binomial",
##      data = data_individ)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5786  -0.9712  -0.7555   1.2398   2.0089
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.336619   0.254014   1.325    0.185
## Alder          -0.045140   0.009185  -4.914 8.91e-07 ***
## KonMan         0.330364   0.135873   2.431   0.015 *
## UtbildareTrafikskola 1.052320   0.157491   6.682 2.36e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1355.9  on 999  degrees of freedom
## Residual deviance: 1271.5  on 996  degrees of freedom
## AIC: 1279.5
##
## Number of Fisher Scoring iterations: 4
```

```
AIC <- -2*logLik(ml_theta,y,X,1000)+2*4
AIC
```

```
## [1] 1279.538
```

The computed AIC, 1279.53, looks similar to the AIC, 1279.5, from the regression model.

```

cross_validate<- 0

for (i in 1:1000){
  new_MLE <- NR(theta0, niter = 3, y[-i], X[-i,])
  loglikelihood <- log(dbinom(y[i], 1, (1+exp(-X[i,] %*% new_MLE))**(-1)))
  cross_validate <- cross_validate + loglikelihood
}

cross_validate_avg <- cross_validate / 1000
cross_validate_avg

```

```
## [1] -0.6397912
```

```

approx_cross_avg <- -AIC/2000
approx_cross_avg

```

```
## [1] -0.6397692
```

We can observe that the cross validated average is quite similar to the mean of our AIC value.

Task 2

In this part we are going to write a code for the posterior density.

```

post <- function(theta, y, X){
  prior <- exp((-1/2)*t(theta) %*% diag((1/100), nrow = 4, ncol =4) %*% theta)
  posteri <- prior * L(theta, y, X, length(y))
}

#Xtest <- cbind(1, 18:25, rep(c(0, 1), 4), rep(c(1, 1, 0, 0), 2))
#ytest <- c(rep(TRUE, 4), rep(FALSE, 4))
#post(c(260, -10, 10, -20), ytest, Xtest) / post(c(270, -15, 15, -25), ytest, Xtest)

```

Task 3

In the last part we are going to implement a metropolis - hasting algorithm that given a starting value, simulates from the posterior distribution of θ . We are going to use the algorithm to sample (at least) 10000 vectors θ from the posterior. A suitable choice of sigma (the step-size) is the vector of ML-estimator standard errors. After that we are going to plot the draws for each parameter, plot histograms of the parameters posteriors, then use the samples to approximate posterior means and 95% credibility intervals (using 'quantile') for the parameters. Compare with the results of the frequentist analysis and lastly find the probability of a pass for someone given my age and sex.

```

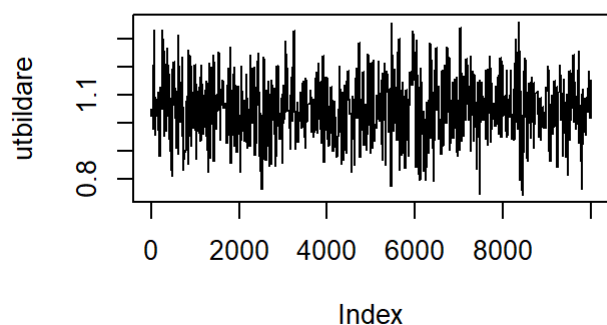
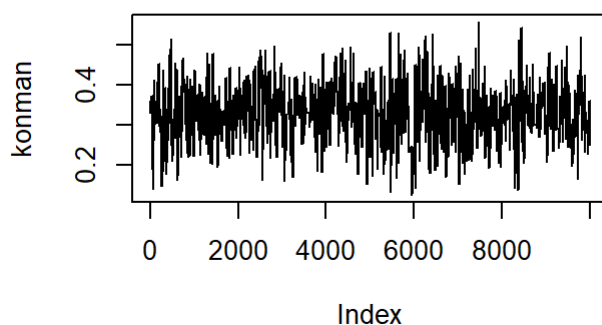
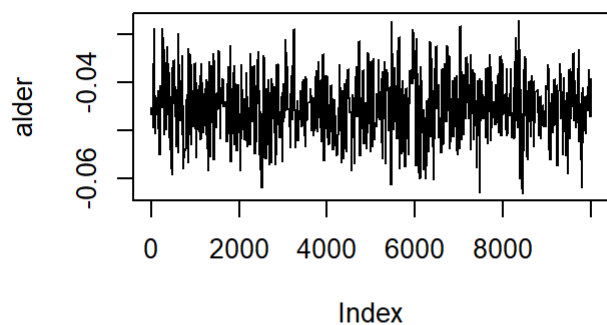
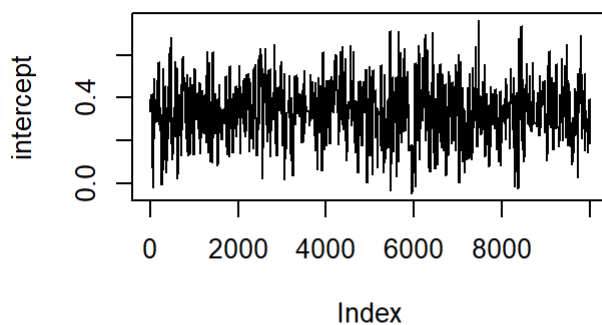
#post <- function(theta){L.weib(theta[1], theta[2], data = ac)}
N <- 10000 #Antal iterationer
theta <- matrix(nrow = N, ncol = 4) #number of col = 4
theta[1,] <- theta_est #Startvärde
sigma <- c(summary(modell)$coefficient[,2]) #Steglängd
for (i in 2:N){
  theta.star <- theta[i-1,] + rnorm(2) * sigma
  if (post(c(theta.star),y,X) / post(c(theta[i-1,]),y,X) > runif(1)){ #using our post function
    theta[i,] <- theta.star
  }
  else{
    theta[i,] <- theta[i-1,]
  }
}

```

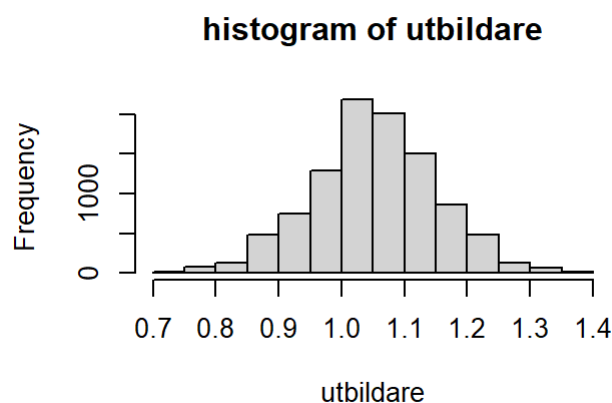
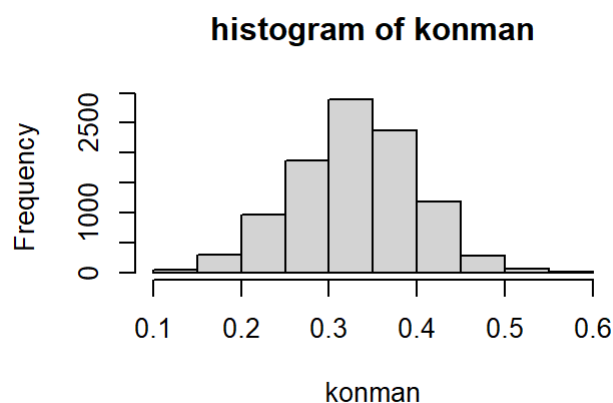
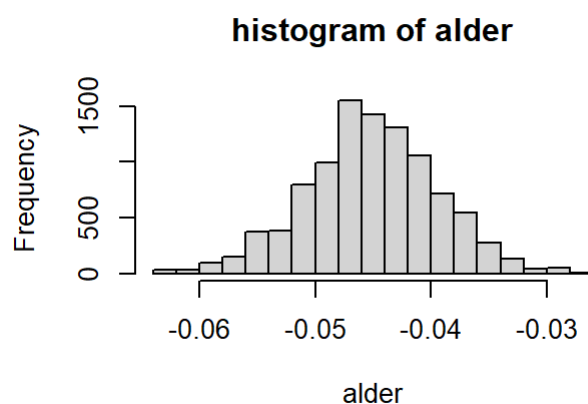
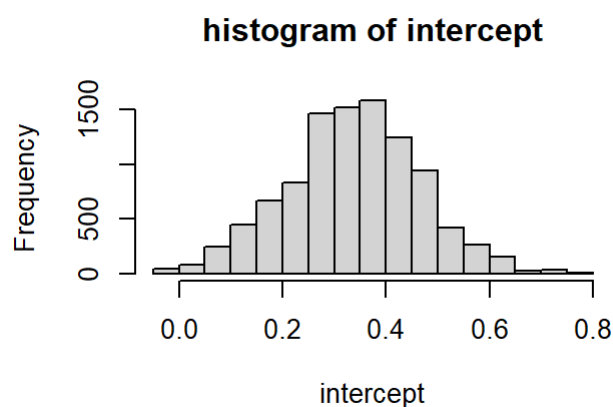
```

par(mfrow=c(2,2))
plot(theta[,1], type = "l", ylab = "intercept")
plot(theta[,2], type = "l", ylab = "alder")
plot(theta[,3], type = "l", ylab = "konman")
plot(theta[,4], type = "l", ylab = "utbildare")

```



```
par(mfrow=c(2,2))
hist(theta[,1],xlab = "intercept", main = "histogram of intercept")
hist(theta[,2], xlab = "alder", main = "histogram of alder")
hist(theta[,3], xlab = "konman", main = "histogram of konman")
hist(theta[,4], xlab = "utbildare", main = "histogram of utbildare")
```



```
posterior_mean <- c(mean(theta[,1]), mean(theta[,2]), mean(theta[,3]), mean(theta[,4]))
posterior_mean
```

```
## [1] 0.33710067 -0.04514484 0.33062298 1.05222648
```

```
theta_est
```

```
## [1] 0.33661607 -0.04513943 0.33036377 1.05231937
```

The results obtained from the posterior mean are indeed quite similar to those from the frequentist analysis.

```
x <- c(1, 26, 0, 0)
p <- 1 / (1+exp(-x %*% posterior_mean))
p
```

```
##           [,1]
## [1,] 0.3022376
```

The approximate of $P(Y=1|y)$ given my sex and age is equal to 30%